NWS CHANGE FORM
PART A

1. DATE SUBMITTED
29 March 2000

This form is in three parts. Submitters must complete unshaded blocks in Part A and as much of Part B as possible. WSH will complete Part C (implementation details). If there is no specific required change date, enter 60 days from date submitted. Address questions to NWS Change Management at (301) 713-1373. Submit change requests to the NWSRC mailbox (External: NWSRC@noaa.gov).

2. ORIGINATOR OFFICE
APO

3. SUBMITTING AUTHORITY
Name: Ward Seguin
Routing Code: W/APO1

4. COGNIZANT TECHNICAL INDIVIDUAL
Name: Carl Bullock
Routing Code:: FSL
Phone:(303) 497-7247

5. ORIGINATOR TRACKING NUMBER
P426_FSL_A100052

6. SYSTEMS AFFECTED BY CHANGE
□ DATA PRODUCTS (Complete Data Products Supplement)
□ ASOS ☒ AWIPS □ CRS □ NEXRAD □OTHER (specify) ________

7. WSH TRACKING NUMBER
NWS 573

8. TITLE OF CHANGE
Fix Mosaic Z problems - Product is slow to load

9. TYPE OF CHANGE
□ HARDWARE ☒ SOFTWARE □ DOCUMENTATION ONLY

10. SITES AFFECTED (Attach Part B, Page 2, if needed)
To be tested at Norman and Boulder, but will be later be pushed by the SST to all 4.2.6 sites.

11. STATEMENT OF REQUIREMENT, PROBLEM, OR DEFICIENCY OF EXISTING SYSTEM (Include problem report reference numbers.)

The Mosaic Z product is VERY slow to load and will consume large amounts of WS resources. When users run a 'top' command on a WS using the product in an 8 frame loop , they often find the SIZE is > 300M and CPU% 30% and higher, with small idle times.. [Note: it is NOT a problem in 4.3]

12. KNOWN OR PROPOSED SOLUTION (Include source and description of new features or data products.)

Make the updated script available to the 4.2.6 sites and get them to regenerate their mosaics by relocalizing. FSL would provide the revised version of the doMosaicProcessing.ksh script. See Attachment A

13. ALTERNATE SOLUTIONS

None

14. REQUIRED CHANGE DATE
ASAP

15. RATIONALE FOR REQUIRED CHANGE DATE (Include proposed priority, if known.)
All sites need to ready for this years severe weather season.

CCB/PMC/CMB DECISION

16. DECISION AUTHORITY LEVEL
□ CCB LEVEL ONLY
□ PMC or NWS CMB DECISION REQUIRED

17. CCB LEVEL DECISION
□ APPROVED
□ RECOMMEND APPROVAL
□ DISAPPROVED
SIGNATURE
DATE SIGNED

FOR USE ONLY WHEN PMC or NWS CMB DECISION REQUIRED

18. PMC OR NWS CMB DECISION
□ APPROVED
□ DISAPPROVED
SIGNATURE
DATE SIGNED

NWS CHANGE FORM **PART B**	1. ORIGINATOR TRACKING NUMBER P426_FSL_A100052
All RC/ECP submissions must also address the following information. Indicate if any areas are unknown or do not apply. State why information is unknown and when it will be available. Attach extra pages if necessary, referencing each applicable subject.	2. WSH TRACKING NUMBER

FUNDING INFORMATION

Estimate costs and indicate known sources of funding. (Include travel time, installation time, administrative time, and software development time when applicable.)	3. SOURCE OF FUNDING	4. TOTAL COST $
5. DEVELOPMENT COSTS (Estimate development costs) 4 hours.	KMOD _____ BASE	AMOUNT
6. OPERATIONAL TEST AND EVALUATION COSTS (Estimate test and evaluation costs) 4 hours	BASE	AMOUNT $0
7. PRODUCTION COSTS (Include acquisition, kit proofing, spares, delivery, and documentation costs)	KMOD _____ BASE	AMOUNT $0
8. COMMUNICATIONS SERVICE/CIRCUITS COSTS (Include installation and recurring costs) N/A		AMOUNT $0
9. IMPLEMENTATION SUPPORT COSTS (Include travel, installation, and administrative costs) TheSST will push the change to the sites	KMOD _____ BASE	AMOUNT $
9A. LIFE CYCLE SUPPORT COSTS (Less communications service/circuits) Contractor (PRC) - NCF Operations	KMOD _____	AMOUNT Unknown

SUPPORTING INFORMATION AND SCHEDULES

Provide detailed information needed to implement the requested change.

10. DEVELOPMENT STATUS/SCHEDULE (Major milestones such as Start, Beta Test, and OT&E) 2 hours each at Boulder and and Norman	11. PRODUCTION STATUS/SCHEDULE (Major milestones such as Solicitation, Contract Start Date, Delivery Date, Kit Proofing, etc.) N/A
12. IMPLEMENTATION/RETROFIT SCHEDULE N/A	13. FACILITY INFORMATION (Attach facility drawings/plans.) N/A
14. COMMUNICATIONS INSTALLED (Type required, who will order, and associated hardware required; attach Part B, Page 2, if needed.) N/A	15. COMMUNICATIONS SERVICE/CIRCUITS TO BE REMOVED N/A
16. REQUIRED CLEARANCES, WAIVERS, AND LICENSES (Include person or organization responsible for obtaining each) N/A	17. COORDINATION OF CHANGE WITH OTHER CHANGES Requires AWIPS SW Ver 4.2.6 to be installed prior to implementation
18. PHYSICAL ITEMS AND DOCUMENTS AFFECTED (Include part, serial, and document numbers. Attach Part B, Page 2, if needed.) N/A	19. STAFF RESOURCE IMPACTS (Skills and workload impact on maintainers, operators, and managers.) No recurring workload impacts.
20. LOGISTICS IMPACTS (Include facilities, maintenance, training, and support equipment impacts.) N/A	21. OPERATIONAL IMPACTS (Include continuity and back up needs and plans.) N/A
22. ADDITIONAL MAJOR CHANGE ACTIVITIES (Include who will accomplish each of them and staff hours required.) N/A	

NWS CHANGE FORM

PART B - PHYSICAL ITEM AND DOCUMENT IMPACT MATRIX SUPPLEMENT

1. ORIGINATOR TRACKING NUMBER

P426_FSL_A100052

This information is required prior to publication of Engineering Modification Notes and Software Release Notes. List physical items to be replaced and specify any changes in related documentation. (Submitters should complete this information, if known. WSH will assist.)

2. WSH TRACKING NUMBER
NWS 573

3. ITEM NAME, CIRCUIT TYPE, SOFTWARE VERSION, OR SITE LOCATION	4. REMOVE REPLACE MODIFY	5. SUPERSEDED ITEM OR CONFIGURATION		6. SUPERSEDING PART NUMBER OR NEW CONFIGURATION	7. DOC TYPE	8. SUPERSEDED DOCUMENT		9. SUPERSEDING DOCUMENT	
		A. PART	B. SERIAL NUMBER(S) OR			A.	B.	A.	B. REV
None									

NWS CHANGE FORM
PART C

1. ORIGINATOR TRACKING NUMBER

P426_FSL_A100052

WSH is responsible for Part C, but submitters may complete sections that would help clarify the change requirement or the necessary implementation actions.

2. WSH TRACKING NUMBER

NWS 573

3. CCB COST EVALUATION

NWS COST   FAA COST $ DOD COST $ OTHER AGENCY COST $ (SPECIFY)_____________   TOTAL COST $

4. IMPLEMENTATION DOCUMENTS REQUIRED

□ Engineering Modification Note   □ Software Release Notes   □ Other Document (Specify)___________

ADDITIONAL IMPLEMENTATION INSTRUCTIONS (e.g., Implementation schedule, parts shipping instructions, equipment disposal procedures, additional documentation required, and status reporting instructions.) Include documentation, data input, notification vehicle, or specific action step required to verify completion of the implementation activity.

5. IMPLEMENTATION ACTIVITY REQUIRED	**6. REQUIRED COMPLETION DATE**	**7. RESPONSIBLE PERSON AND OFFICE**	**8. DOCUMENT OR ACTION REQUIRED TO VERIFY COMPLETION**
A. Coordinate implementation schedule with sites through SST	30 Mar 00	Thigpen/SST, W/APO3	N/A
B. NCF Pushes the chage to sites as directed by SST	3 April 00	Thigpen/SST, W/APO3	
C. Ensure the appropriate WSH management information systems and configuration management data bases are updated to reflect these changes	2 June 00	Michelle deTommaso W/OSO113	N/A

Attachment A

Author: James E Ramer at MAILHUB
Date: 3/9/00 6:53 PM
Priority: Normal
TO: Jagdish Sharma at W-APO
Subject: Mosaic Z Problems
------------------------------- Message Contents -------------------------------
What is needed is to install a new version of the script file
doMosaicProcessing.ksh, which lives in /awips/fxa/data/localization/scripts.

After installing the new doMosaicProcessing.ksh, on all nodes, the sites
need to run the -radar localization task on all nodes.

This only applies to 4.2, 4.3 onward should be OK.

Here is the new version of doMosaicProcessing.ksh.

==

```
#!/bin/ksh
# ----------------------------------------------------------------------------
# This software is in the public domain, furnished "as is", without technical
# support, and with no warranty, express or implied, as to its usefulness for
# any purpose.
#
# doMosaicProcessing.ksh
# add radar mosaic metadata if appropriate.
#
# Author: Jim Ramer
# ----------------------------------------------------------------------------

. ${FXA_LOCALIZATION_SCRIPTS}/ksh.setUp

# determine number of radars, and number of scales ...exit if we can't
if
   [ -s radarsOnMenu.txt ]
then
   nradars=`wc radarsOnMenu.txt | cut -f 1 "-d "`
else
   exit
fi
if
   [ -s radarDataMenus.txt ]
then
   nded=`grep '^menu:' radarDataMenus.txt | wc | cut -f 1 "-d "`
else
   nded=0
fi
scaleInfo=`$getPath ./ $data_path scaleInfo.txt`
if
   [ -s $scaleInfo ]
then
   nscales=`wc $scaleInfo | cut -f 1 "-d "`
else
   exit
fi

# change single dedicated radar identifier into "Radar"
#if
#   [ $nded -lt 2 ]
#then
#   rm temp1 2 /dev/null
#   radar=`grep "# begin" radarDataMenus.txt |cut -f 3 "-d "`
#   sed 's/"'$radar'"/"Radar"/g' radarDataMenus.txt  temp1
#   if
#     [ -s temp1 ]
#   then
#      mv temp1 radarDataMenus.txt
#   fi
#fi

# If no mosiacs, then we are done
if
   [ -s mosaicScales.txt ]
then
   mscales=`wc mosaicScales.txt | cut -f 2 "-d "`
else
   exit
fi
if
   [ $nradars -lt 2 -o $mscales -eq 0 ]
then
   exit
fi

echo "yes, include radar mosaics"

# create scale assignment mask for product button keys
i=0
scaleMap=" "
while [ $i -lt $nscales ]
do
   if
     grep $i mosaicScales.txt 2 /dev/null
   then
     scaleMap="$scaleMap 1"
   else
     scaleMap="$scaleMap 0"
   fi
   i=`expr $i + 1`
done

# do scale assignments for product button keys
prodButtons=`$getPath $data_path mosaicProductButtons.template`
cat $prodButtons |
while read line
do
   lead=`echo $line | cut -f 1 "-d@"`
   trail=`echo $line | cut -f 2 "-d@"`
   button=`echo $line | cut -f 1 "-d|" | cut -f 1 "-d "`
   comma=FALSE
   for column in $scaleMap
   do
      if
        [ $comma = TRUE ]
      then
        lead="$lead, "
      fi
      if
        [ $column = 1 ]
      then
        lead="${lead}$button"
      else
        lead="${lead}0"
      fi
      comma=TRUE
   done
   echo "${lead}$trail"  radarProductButtons.txt
done

# generate subset of depict keys that are only those on the menu
```

```
rm menuKeys.tmp 2 /dev/null
radarList=`cat radarsOnMenu.txt | cut "-d " -f 1`
for oneRadar in $radarList
  do
  grep $oneRadar radarDepictKeys.txt  menuKeys.tmp
  done
if
    [ -s menuKeys.tmp ]
then
    touch menuKeys.tmp
else
    exit
fi

PUP_TABLES=`$configValue PUP_TABLES mainConfig.txt FALSE`

if [ "$PUP_TABLES" = "FALSE" ]
then
   defZ=20
   def3=21
   defV=19
   defV3=39
   defP=20
   defW=21
   defS=50
   def8=35
   defH=40
else
   defZ=43
   def3=48
   defV=45
   defV3=46
   defP=47
   defW=21
   defS=50
   def8=35
   defH=40
fi

# 16, 8 level reflectivity
RADAR_Z=`$configValue RADAR_Z mainConfig.txt $defZ`
RADAR_3=`$configValue RADAR_3 mainConfig.txt $def3`

# 16, 8 level velocity
RADAR_V=`$configValue RADAR_V mainConfig.txt $defV`
RADAR_V3=`$configValue RADAR_V3 mainConfig.txt $defV3`

# precip
RADAR_P=`$configValue RADAR_P mainConfig.txt $defP`

# spectrum width
RADAR_W=`$configValue RADAR_W mainConfig.txt $defW`

# radial shear
RADAR_S=`$configValue RADAR_S mainConfig.txt $defS`

# 8 bit full scale reflectivity
RADAR_8=`$configValue RADAR_8 mainConfig.txt $def8`

# 8 bit hybrid scan reflectivity
RADAR_H=`$configValue RADAR_H mainConfig.txt $defH`

# generate new radar mosaic depictable keys
rm temp1 temp2 2 /dev/null
depictKeys=`$getPath $data_path mosaicDepictKeys.template`
cat $depictKeys | grep -v '^//' | \
  sed 's/@Z@/'$RADAR_Z'/g' | \
  sed 's/@3@/'$RADAR_3'/g' | \
  sed 's/@W@/'$RADAR_W'/g' | \
  sed 's/@P@/'$RADAR_P'/g' | \
  sed 's/@S@/'$RADAR_S'/g' | \
  sed 's/@V@/'$RADAR_V'/g' | \
  sed 's/@O@/'$RADAR_V3'/g' | \
  sed 's/@8@/'$RADAR_8'/g' | \
  sed 's/@H@/'$RADAR_H'/g' |
while read line
do
    srchstr=`echo $line| cut "-d|" -f 7`
    keylist=`grep "$srchstr" menuKeys.tmp | cut "-d|" -f 1`
    del=""
    keytext=""
    lastrad=-1
    for key in $keylist
    do
        rad=`expr $key / 65536`
        if
            [ $key -gt 1073741824 -a  $rad -ne $lastrad ]
        then
            keytext="$keytext$del$key"
            del=","
        fi
        lastrad=$rad
    done
    lead=`echo "$line"| cut "-d@" -f 1`
    trail=`echo "$line"| cut "-d@" -f 2`
    echo "$lead$keytext$trail"  radarDepictKeys.txt
done
rm temp1 temp2 menuKeys.tmp 2 /dev/null

# add specified radar menu keys to radarDataMenus.txt
dataMenus=`$getPath $data_path mosaicDataMenus.template`
cat $dataMenus  mosiacDataMenus.txt
```